CLAIMS

1-24. (Canceled)

25. (Previously Presented) A programmable processor for executing a plurality of programs, said programmable processor comprising:

a target program counter coupled to a plurality of program counters, said target program counter for determining a number of programs to interleave from a plurality of programs that is greater than the number of program counters;

each of said plurality of program counters coupled to an instruction memory;

instructions from said instruction memory coupled to an instruction decode;

said decode coupled to a plurality of registers;

each of said plurality of registers coupled to an operand route;

said operand route coupled to an arithmetic datapath;

said datapath and an output of a data memory coupled to a result route; and

an output of said result route fed back to each of said plurality of registers.

26. (Previously Presented) The programmable processor of claim 25 wherein said plurality of program counters is equal to said plurality of programs to be interleaved.

27. (Previously Presented) The programmable processor of claim 25 wherein said plurality of registers is equal to said plurality of programs to be interleaved.

28. (Previously Presented) The programmable processor of claim 25 wherein said plurality of registers is more than said plurality of programs to be interleaved.

29. (Previously Presented) The programmable processor of claim 25 wherein said instruction memory is larger than needed to hold said plurality of programs.

30. (Previously Presented) The programmable processor of claim 25 wherein said data memory is larger than needed to hold a data set accessed by each of said plurality of programs.

31. (Previously Presented) The programmable processor of claim 25 wherein each of said plurality of registers is double buffered and contains twice as many copies of registers as said plurality of programs.

32. (Canceled)

33. (Currently Amended) A method, by a programmable processor, of executing one or more instructions from a plurality of programs, comprising:

assigning a first output register slot to a first of said plurality of programs wherein each of the plurality of programs comprises a plurality of instructions;

executing said one or more instructions of said first program until said first program is completed;

loading output of said first program into its reserved space when said first program is completed;

checking to see if all of said plurality of ~~programs~~ programs, including said first program, are completed;

checking to see if a second output register slot is available to assign to a second program from said plurality of programs when said first program is completed;

checking to see if one or more instructions are available when at least one of said plurality of programs is not completed; and

placing an no-op when no more instructions are available or said second output register slot is not available.